



THUNDERBOLTS AND LIGHTNING
VERY, VERY FRIGHTENING

SNARE & RZN
@ SYSCAN
APRIL 2014

OBLIGATORY INTRO SLIDE

WHO ARE THESE IDIOTS?

- ▶ rzn aka Sam
 - ▶ PhD student at UoA
 - ▶ research into ray-tracing on FPGAs
 - ▶ extensive collection of name tags and hair nets
- ▶ snare aka Loukas
 - ▶ computer guy at Azimuth Security
 - ▶ did some OS X kernel and UEFI firmware stuff one time
 - ▶ world's strongest millionaire
 - ▶ internet-famous feet

Thunderbolts and Lightning ⚡⚡⚡ Very, Very Frightenin



WHAT IS THIS TALK ABOUT?

- ▶ Apparently Thunderbolt DMA attacks are totally a thing
- ▶ But we haven't seen a PoC yet
- ▶ And it sounded like fun
- ▶ It's not actually about Lightning (the iDevice connector)
 - ▶ Sorry Stefan

AGENDA

THINGS WHAT WE IS GOING TO TALK ABOUT

- ▶ FireWire DMA attacks
- ▶ Thunderbolt
- ▶ How is PCIe formed?
- ▶ What the fuck is an FPGA?
- ▶ Our approach to attacking Thunderbolt
- ▶ Sweet stunt hack demo and stuff
- ▶ Defence

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

HIT BY THE SHORT BUS

FIREWIRE DMA ATTACKS

- ▶ See Metlstorm's "Hit By A Bus" circa 2006 (Ruxcon)
 - ▶ First done by Quinn the Eskimo (Apple awesome dude)
 - ▶ Won MacHack 2002 by drawing a screensaver over FireWire!
 - ▶ See also Inception - a FireWire DMA tool
- ▶ How does it work?
 - ▶ Using SBP-2
 - ▶ Firewire chipset does DMA R/W on PCIe bus
 - ▶ Stream data out FW interface

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

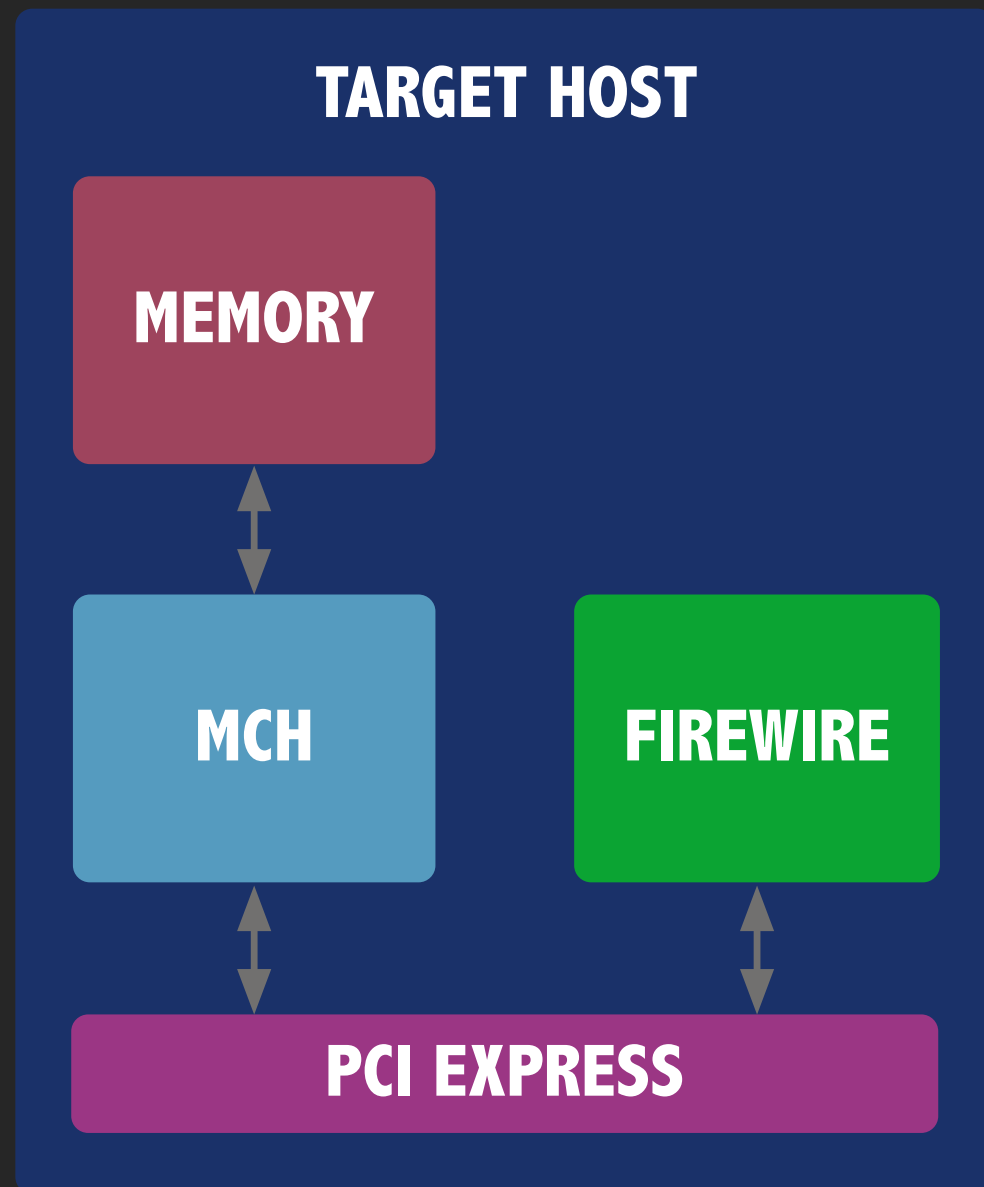
HIT BY THE SHORT BUS

FIREWIRE DMA ATTACKS

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

HIT BY THE SHORT BUS

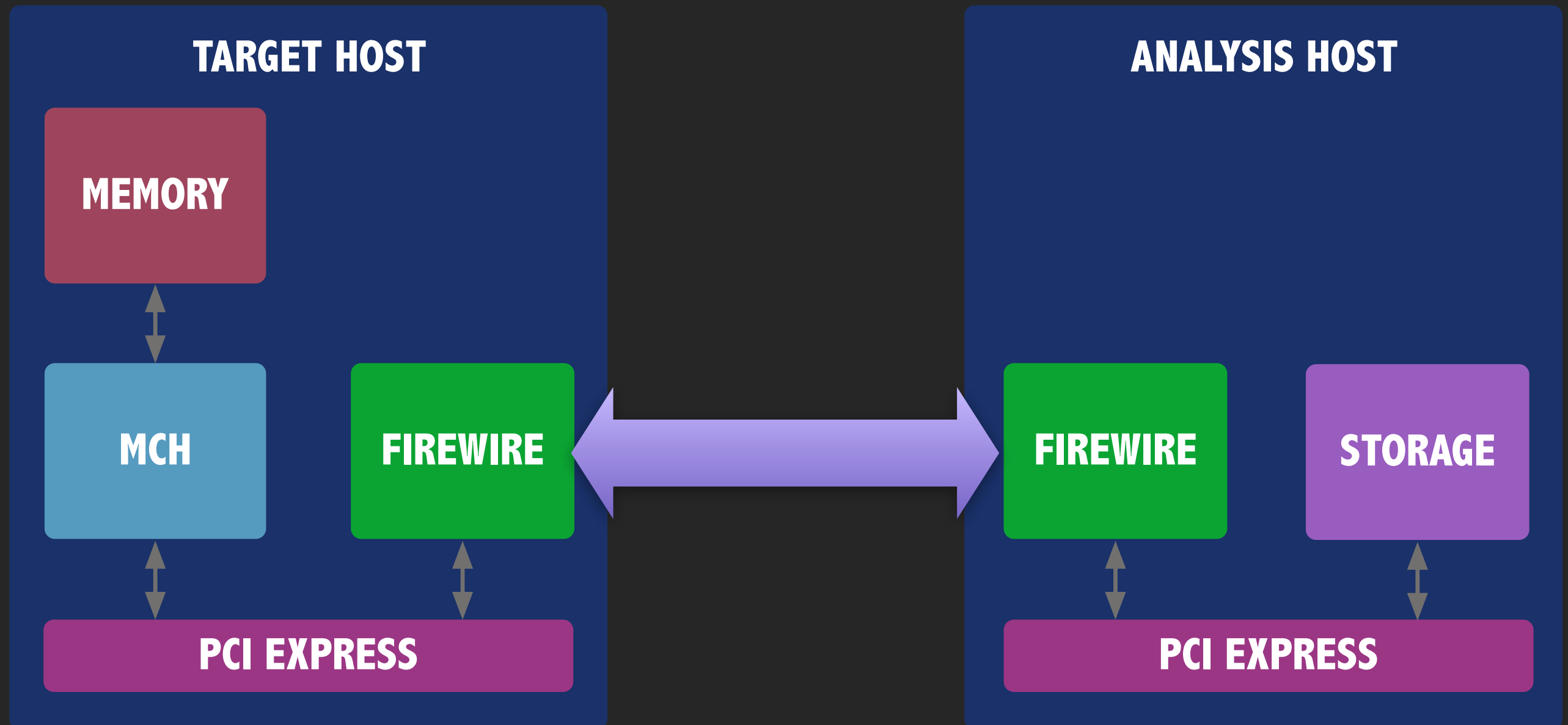
FIREWIRE DMA ATTACKS



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

HIT BY THE SHORT BUS

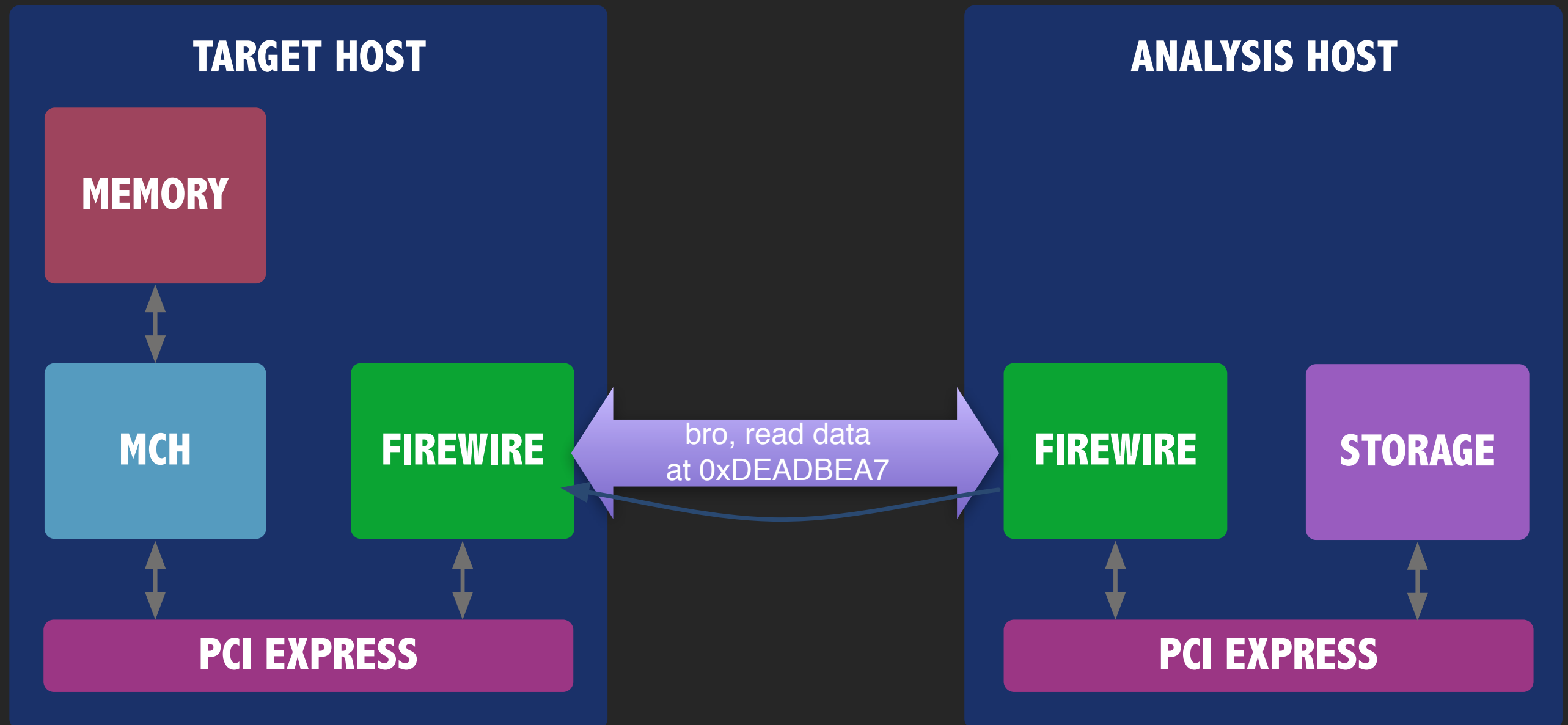
FIREWIRE DMA ATTACKS



Thunderbolts and Lightning ⚡⚡⚡ Very, Very Frightening

HIT BY THE SHORT BUS

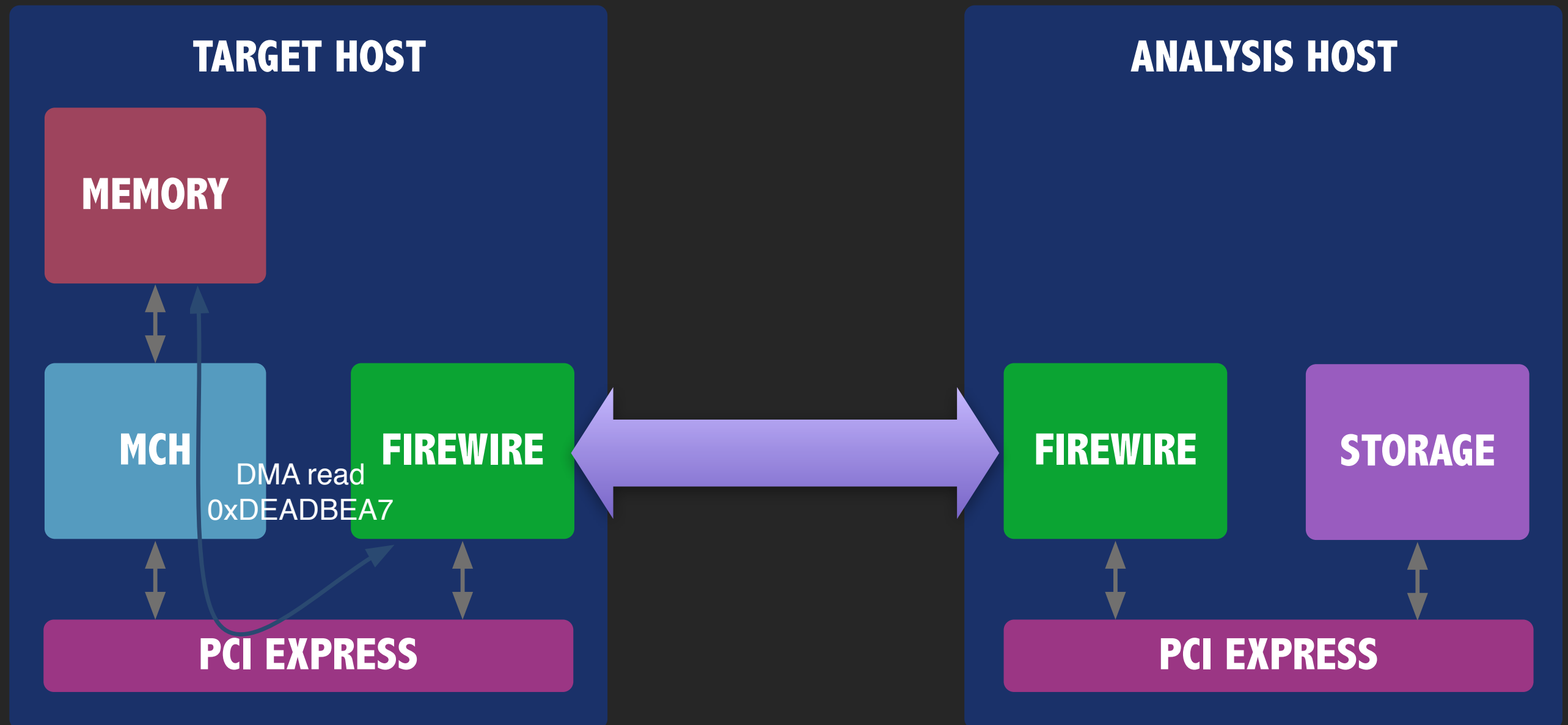
FIREWIRE DMA ATTACKS



Thunderbolts and Lightning ⚡⚡⚡ Very, Very Frightening

HIT BY THE SHORT BUS

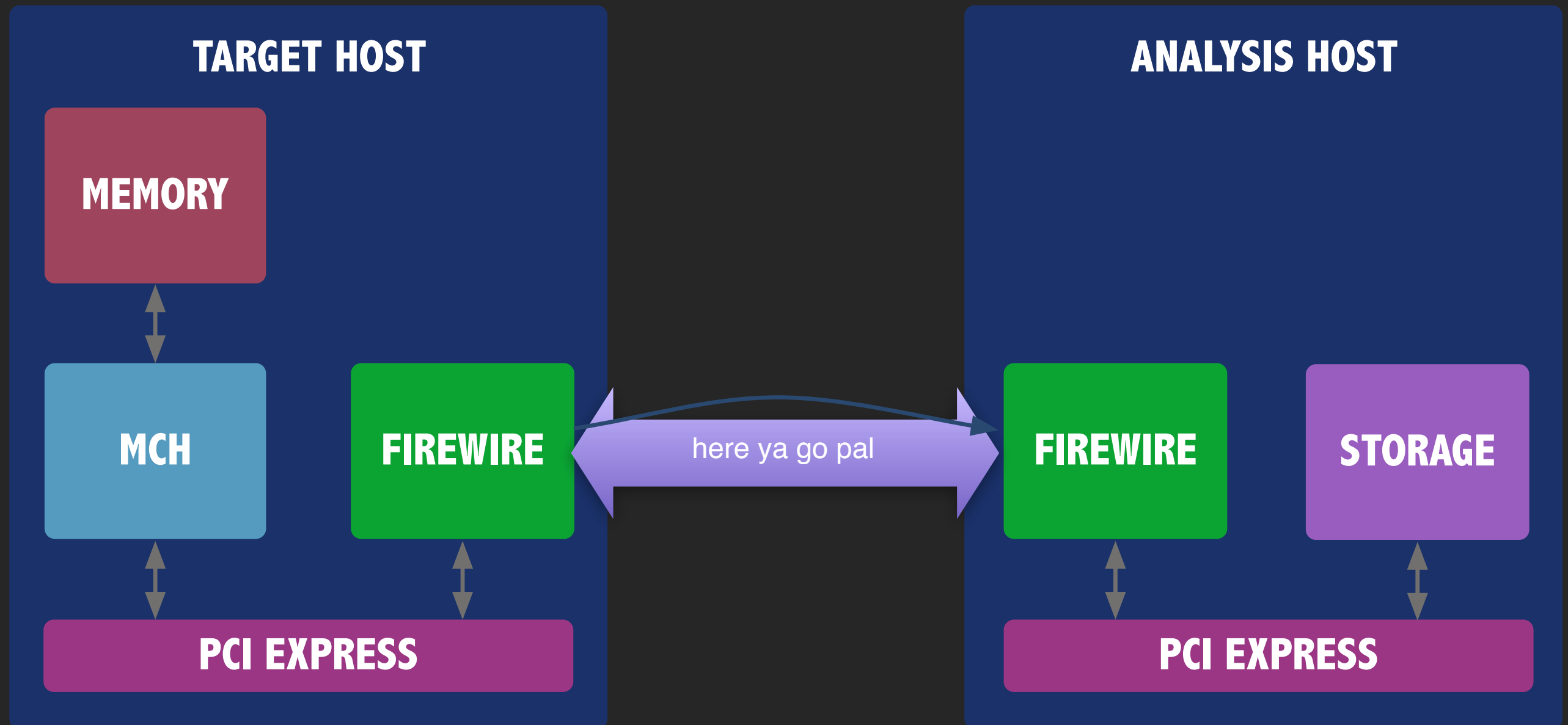
FIREWIRE DMA ATTACKS



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

HIT BY THE SHORT BUS

FIREWIRE DMA ATTACKS



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

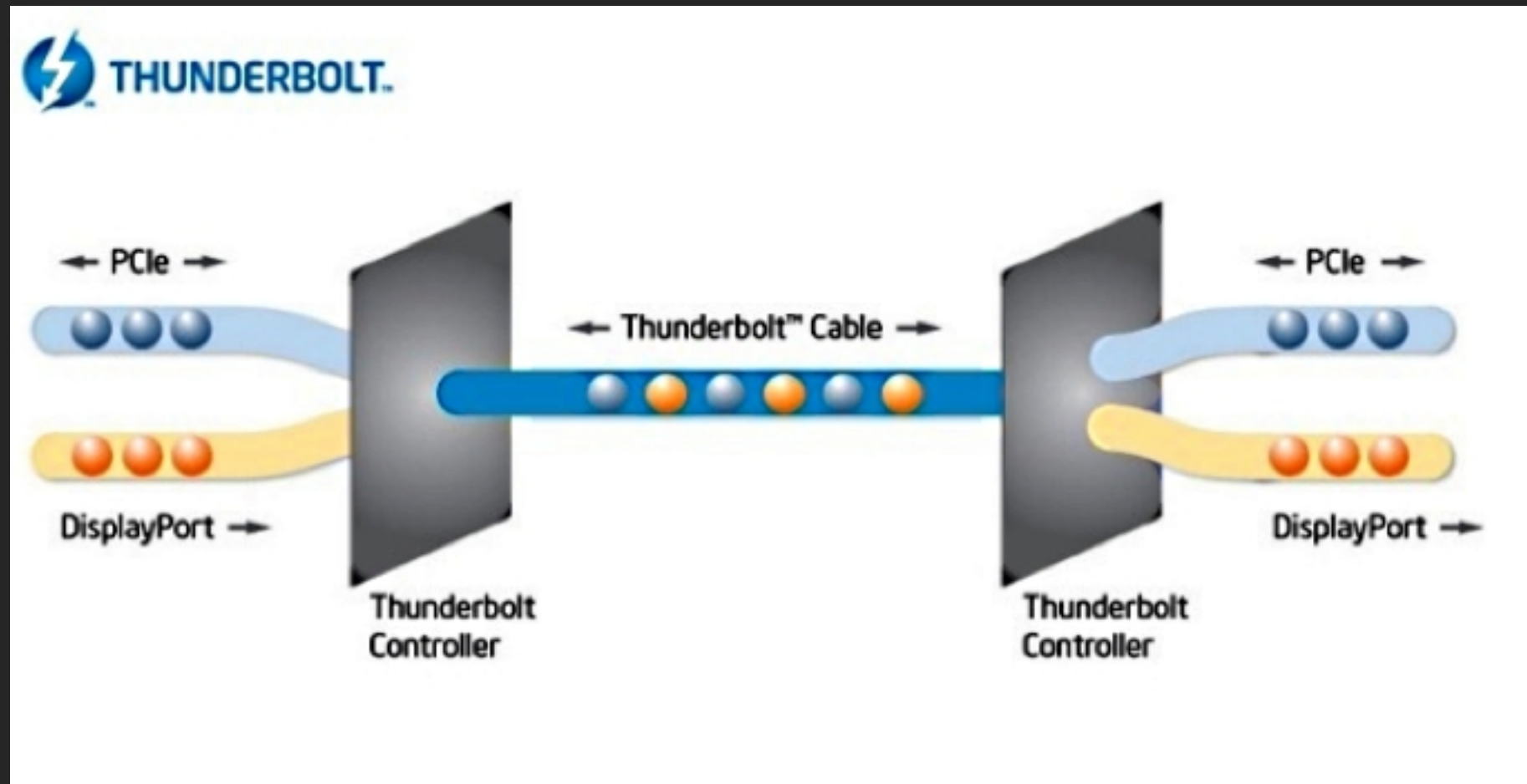
HIT BY THE SHORT BUS

LIMITATIONS

- ▶ Obviously requires that there be a FireWire interface
- ▶ 32-bit addressing = only lower 4GB of RAM
- ▶ On OS X FireWire DMA is disabled when the screen is locked & FileVault is enabled
 - ▶ Kernel tells FW chipset not to do DMA any more
 - ▶ #sadface

WHAT'S A THUNDERBOLT? EH?

- ▶ Thunderbolt == PCIe + DisplayPort + pixie dust

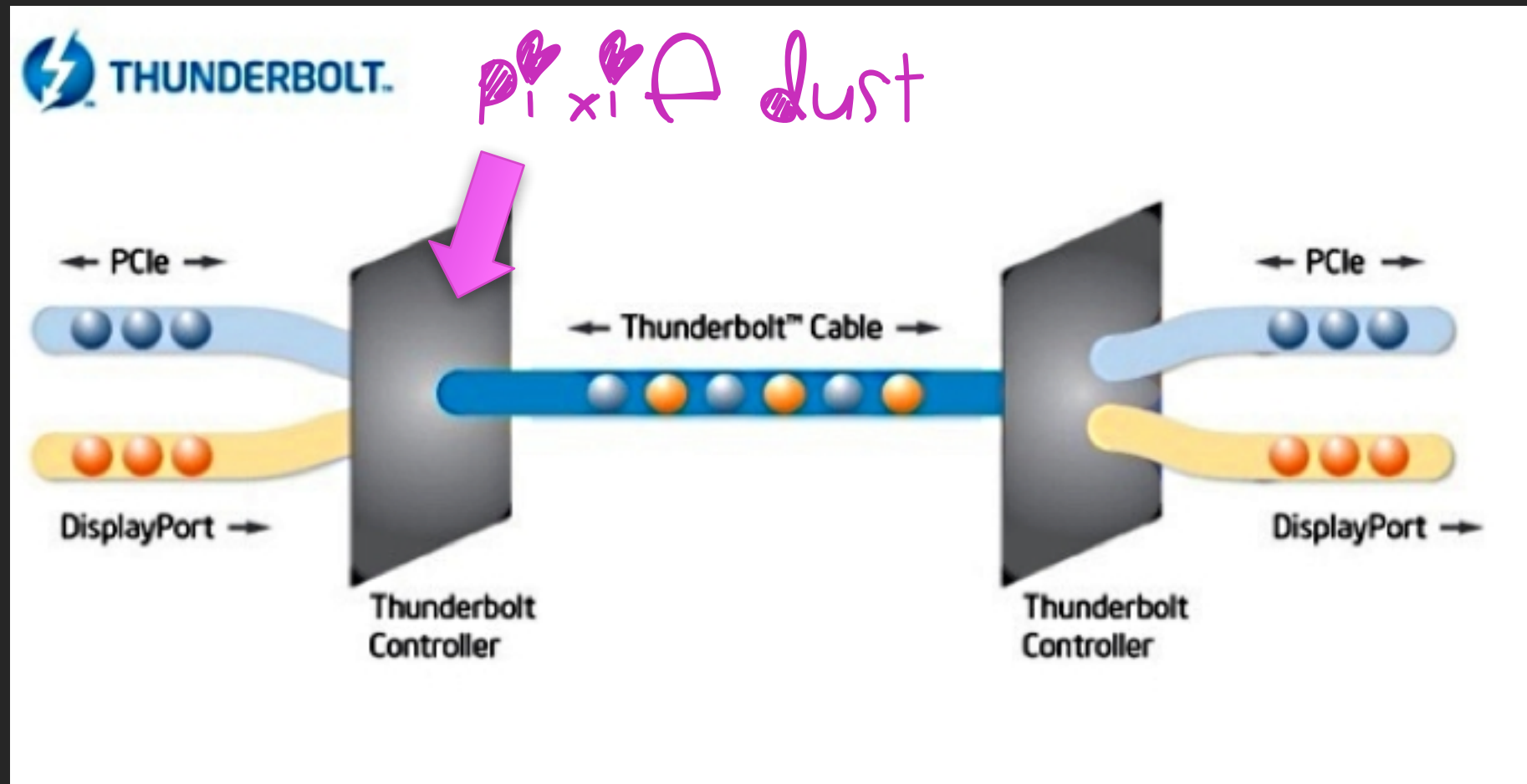


- ▶ Send DMA requests directly over PCIe?

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

WHAT'S A THUNDERBOLT? EH?

- ▶ Thunderbolt == PCIe + DisplayPort + pixie dust



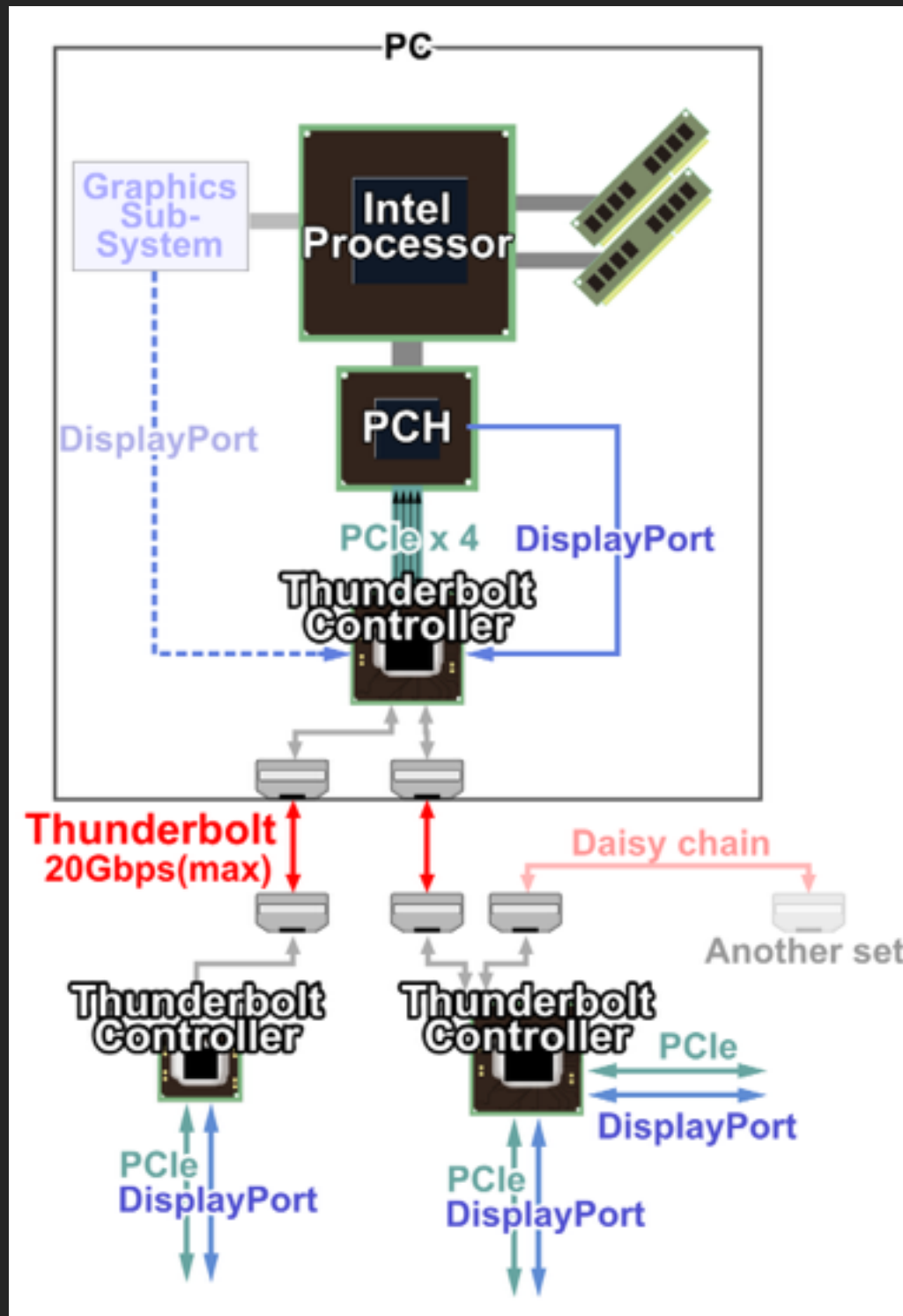
- ▶ Send DMA requests directly over PCIe?

Thunderbolts and Lightning ⚡⚡⚡ Very, Very Frightening

WHAT'S A THUNDERBOLT?

PICS OR GTFO

Slightly more
useful diagram



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

THUNDERBOLT DMA THUS FAR

CHEATING WITH FIREWIRE

- ▶ “Thunderbolt DMA”
 - ▶ Connect Thunderbolt to FireWire adapter
 - ▶ ???
 - ▶ Profit
- ▶ Subject to the same limitations as regular FireWire

HOW IS PCIE FORMED?

WHEN PCI AND PCI-X LOVE EACH OTHER VERY MUCH

► Serial point-to-point interconnect

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

HOW IS PCIE FORMED?

WHEN PCI AND PCI-X LOVE EACH OTHER VERY MUCH

- ▶ Serial point-to-point interconnect
- ▶ A lane consists of a tx and rx differential pair
(4 wires per lane)

HOW IS PCIE FORMED?

WHEN PCI AND PCI-X LOVE EACH OTHER VERY MUCH

- ▶ Serial point-to-point interconnect
- ▶ A lane consists of a tx and rx differential pair
(4 wires per lane)
- ▶ Scalable number of lanes, negotiated at link setup

HOW IS PCIE FORMED?

WHEN PCI AND PCI-X LOVE EACH OTHER VERY MUCH

- ▶ Serial point-to-point interconnect
- ▶ A lane consists of a tx and rx differential pair
(4 wires per lane)
- ▶ Scalable number of lanes, negotiated at link setup
- ▶ Layered, packet based, transaction protocol
 - ▶ Physical layer
 - ▶ Data link layer
 - ▶ Transaction layer

HOW IS PCIE FORMED?

WHEN PCI AND PCI-X LOVE EACH OTHER VERY MUCH

- ▶ Serial point-to-point interconnect
- ▶ A lane consists of a tx and rx differential pair
(4 wires per lane)
- ▶ Scalable number of lanes, negotiated at link setup
- ▶ Layered, packet based, transaction protocol
 - ▶ Physical layer
 - ▶ Data link layer
 - ▶ Transaction layer
- ▶ Level sensitive or message signaled interrupts

HOW IS PCIE FORMED?

DMA

- ▶ Four transaction types

- ▶ I/O read/write
- ▶ Configuration read/write
- ▶ Memory read/write
- ▶ Messaging

- ▶ DMA:

- ▶ Configuration write to grant device “bus master”
- ▶ Write target address and command to device
- ▶ Device interrupts when finished

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

WTF IS AN FPGA?

[1] WIKIPEDIA

► Field Programmable ~~Gatorade~~ Gate Array

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

WTF IS AN FPGA?

[1] WIKIPEDIA

- ▶ Field Programmable ~~Gaterade~~ Gate Array
- ▶ Matrix of configurable logic blocks, each containing 'slices'

WTF IS AN FPGA?

[1] WIKIPEDIA

- ▶ Field Programmable ~~Gaterade~~ Gate Array
- ▶ Matrix of configurable logic blocks, each containing 'slices'
- ▶ Slice contents are the core of FPGA functionality
 - ▶ Look up tables (LUTs)
 - ▶ Flip-flops
 - ▶ Carry chain
 - ▶ Muxes

WTF IS AN FPGA?

[1] WIKIPEDIA

- ▶ Field Programmable ~~Gaterade~~ Gate Array
- ▶ Matrix of configurable logic blocks, each containing 'slices'
- ▶ Slice contents are the core of FPGA functionality
 - ▶ Look up tables (LUTs)
 - ▶ Flip-flops
 - ▶ Carry chain
 - ▶ Muxes
- ▶ Additional general features: blockRAMs, FIFOs, DSP blocks, clocking resources (PLLs, DCMs)

WTF IS AN FPGA?

[1] WIKIPEDIA

- ▶ Field Programmable ~~Gaterade~~ Gate Array
- ▶ Matrix of configurable logic blocks, each containing 'slices'
- ▶ Slice contents are the core of FPGA functionality
 - ▶ Look up tables (LUTs)
 - ▶ Flip-flops
 - ▶ Carry chain
 - ▶ Muxes
- ▶ Additional general features: blockRAMs, FIFOs, DSP blocks, clocking resources (PLLs, DCMs)
- ▶ Device specific features: PCIe, Ethernet, DDR2/3

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

WTF IS AN FPGA?

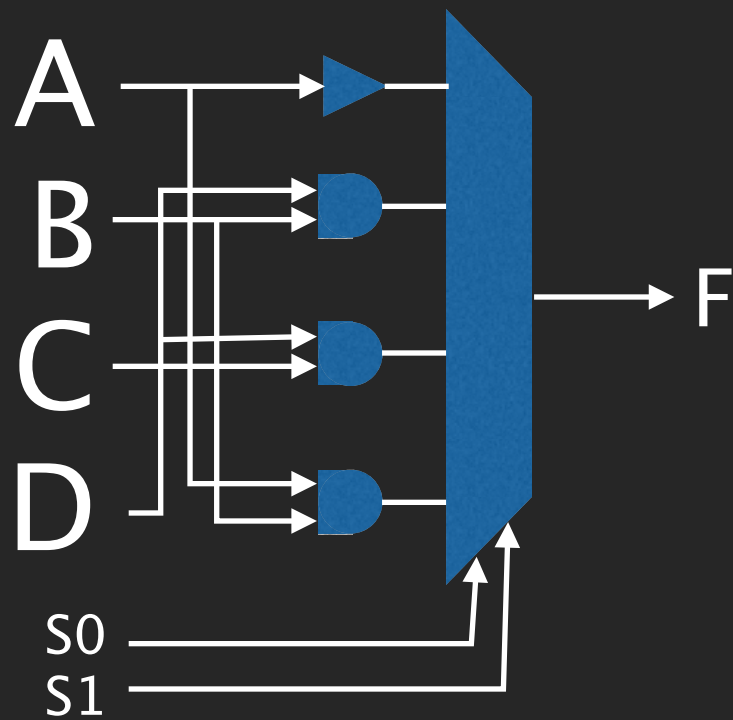
[1] WIKIPEDIA

- ▶ Field Programmable ~~Gaterade~~ Gate Array
- ▶ Matrix of configurable logic blocks, each containing 'slices'
- ▶ Slice contents are the core of FPGA functionality
 - ▶ Look up tables (LUTs)
 - ▶ Flip-flops
 - ▶ Carry chain
 - ▶ Muxes
- ▶ Additional general features: blockRAMs, FIFOs, DSP blocks, clocking resources (PLLs, DCMs)
- ▶ Device specific features: PCIe, Ethernet, DDR2/3
- ▶ Reprogrammable

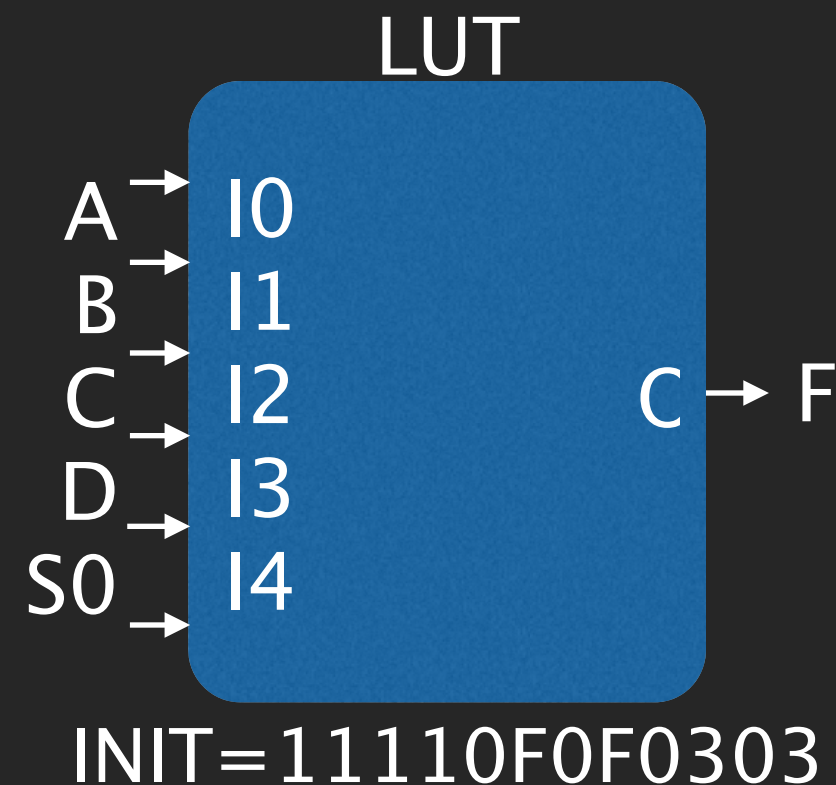
Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

LUTS LUTS LUTS

► logic ► truth table ► LUT



S1	S0	D	C	B	A	F
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	1
⚡	⚡	⚡	⚡	⚡	⚡	⚡
1	1	1	1	0	0	1
1	1	1	1	0	1	0
1	1	1	1	1	0	1
1	1	1	1	1	1	0



- ▶ A LUT is essentially a 6-input memory, containing the desired output for each set of inputs (addresses)
- ▶ It doesn't matter how simple or complex the function, it is only limited by the inputs

Thunderbolts and Lightning ⚡⚡⚡ Very, Very Frightening

WTF IS AN FPGA?

IT'S ALL ABOUT THE LOLS

- ▶ Application logic is described in an HDL; verilog or vhdI

WTF IS AN FPGA?

IT'S ALL ABOUT THE LOLS

- ▶ Application logic is described in an HDL; verilog or vhd
- ▶ You can leave it all to the synthesis tool to infer logic, but it is important to understand how a LUT works

WTF IS AN FPGA?

IT'S ALL ABOUT THE LOLS

- ▶ Application logic is described in an HDL; verilog or vhd
- ▶ You can leave it all to the synthesis tool to infer logic, but it is important to understand how a LUT works
- ▶ Maximum frequency determined by “levels of logic”
 - ▶ A level of logic is the combination of LUT delay and routing delay between two flip-flops
 - ▶ LUT delay = static, constant property of the device
 - ▶ Routing delay = dynamic, influenced by LUT placement

WTF IS AN FPGA?

IT'S ALL ABOUT THE LOLS

- ▶ Application logic is described in an HDL; verilog or vhd
- ▶ You can leave it all to the synthesis tool to infer logic, but it is important to understand how a LUT works
- ▶ Maximum frequency determined by “levels of logic”
 - ▶ A level of logic is the combination of LUT delay and routing delay between two flip-flops
 - ▶ LUT delay = static, constant property of the device
 - ▶ Routing delay = dynamic, influenced by LUT placement
- ▶ Reduce levels of logic, place LUTs closer together = higher clock frequency

420 MicroBlazeIt

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

- ▶ Microblaze is a micro-controller that can be implemented in FPGA logic

420MicroBlazeIt

- ▶ Microblaze is a micro-controller that can be implemented in FPGA logic
- ▶ Interfaces with AXI bus
 - ▶ Standard interface to easily memory map other custom or off-the-shelf IP blocks

420MicroBlazeIt

- ▶ Microblaze is a micro-controller that can be implemented in FPGA logic
- ▶ Interfaces with AXI bus
 - ▶ Standard interface to easily memory map other custom or off-the-shelf IP blocks
- ▶ Code is written in C or C++, compiled with XSDK

420MicroBlazeIt

- ▶ Microblaze is a micro-controller that can be implemented in FPGA logic
- ▶ Interfaces with AXI bus
 - ▶ Standard interface to easily memory map other custom or off-the-shelf IP blocks
- ▶ Code is written in C or C++, compiled with XSDK
- ▶ Really useful for writing control logic
 - ▶ Previously you'd write large state machines in HDL
 - ▶ Also means noobs (snare) can write code for it

420MicroBlazeIt

- ▶ Microblaze is a micro-controller that can be implemented in FPGA logic
- ▶ Interfaces with AXI bus
 - ▶ Standard interface to easily memory map other custom or off-the-shelf IP blocks
- ▶ Code is written in C or C++, compiled with XSDK
- ▶ Really useful for writing control logic
 - ▶ Previously you'd write large state machines in HDL
 - ▶ Also means noobs (snare) can write code for it
- ▶ Connect it via serial and you can printf debug your logic!

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

WTF IS AN FPGA?
BUTT, HOW DO WE DO PCIE?

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

WTF IS AN FPGA?

BUTT, HOW DO WE DO PCIE?

- ▶ AXI PCIE core uses FPGA device specific features to implement PCIE

WTF IS AN FPGA?

BUTT, HOW DO WE DO PCIE?

- ▶ AXI PCIE core uses FPGA device specific features to implement PCIE
- ▶ Memory mapped to MicroBlaze
 - ▶ Read/write to memory mapped AXI core translates to PCIE read/write TLPs
 - ▶ Read/write TLPs from PCIE translate to memory mapped AXI core read/write

OUR APPROACH

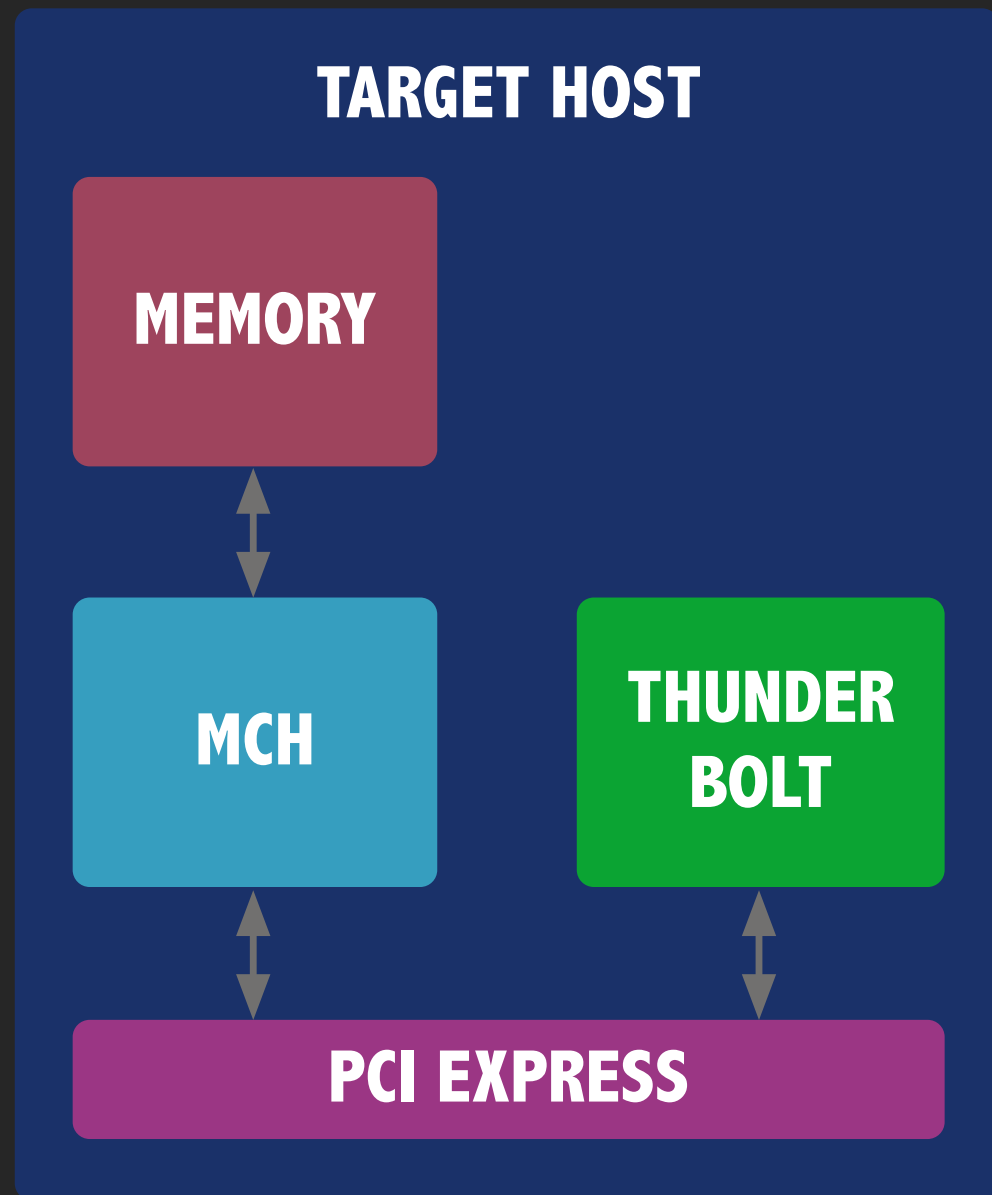
FAKE IT TILL YOU BREAK IT

- ▶ Become bus master
- ▶ ???
- ▶ Profit

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

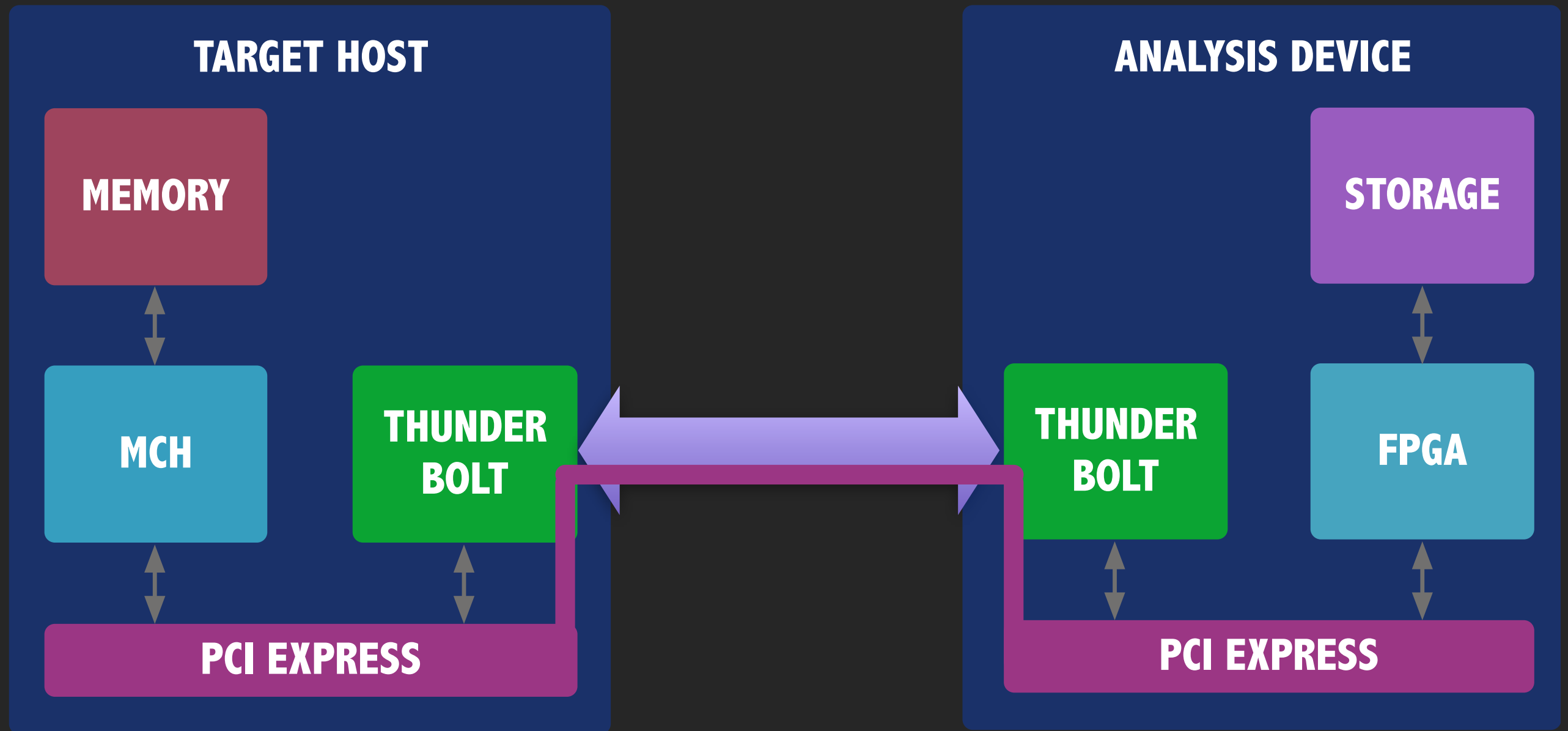
MEMORY CAPTURE

THUNDERBOLT DMA



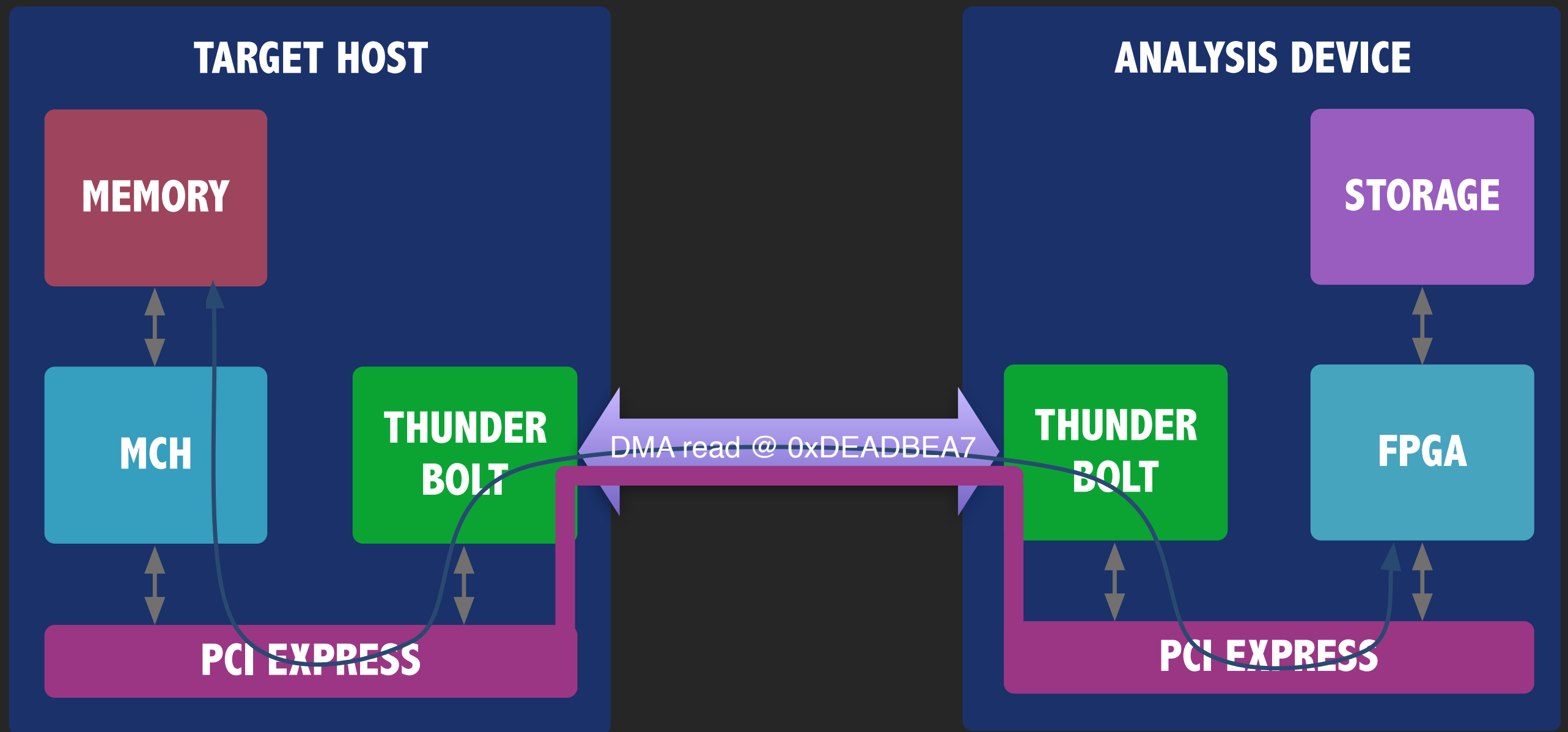
MEMORY CAPTURE

THUNDERBOLT DMA



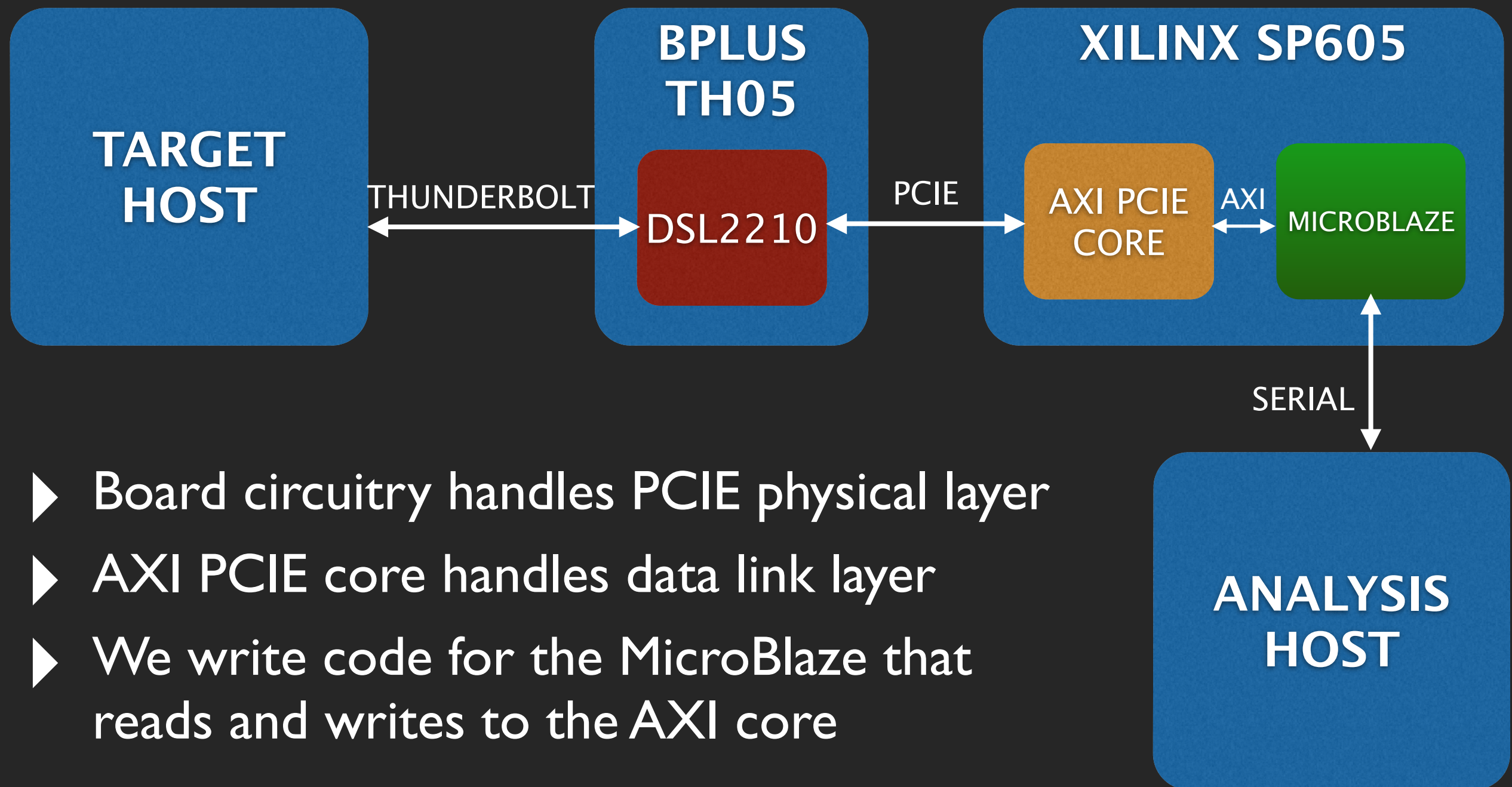
MEMORY CAPTURE

THUNDERBOLT DMA



OUR APPROACH

FAKE IT TILL YOU BREAK IT



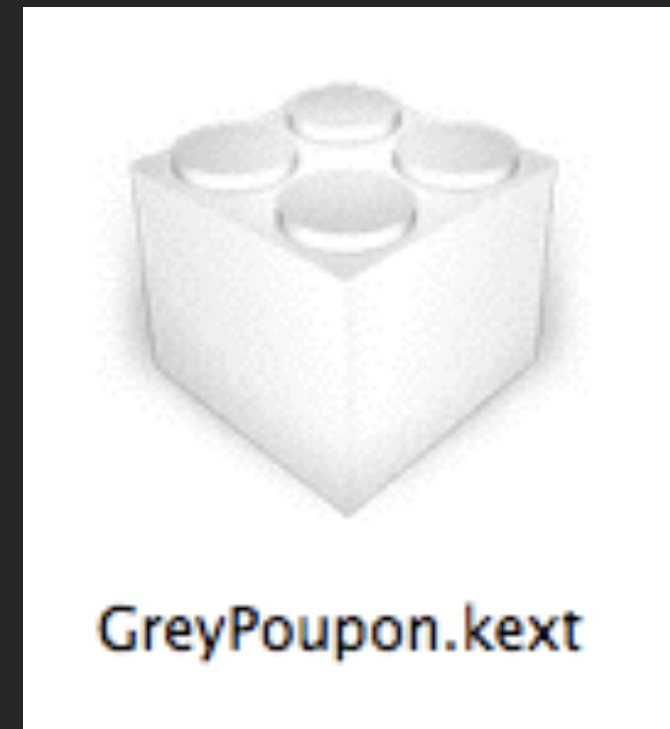
Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

ATTACKING A MAC

OK, SO FPGA TALKS PCIe

- ▶ Phase 1 - write our own driver

- ▶ Make FPGA bus master
- ▶ Tell it what to do



- ▶ Phase 2 - imitate another device

- ▶ Change device id, vendor id in configuration space
- ▶ Trick the OS into loading an existing driver that will make us bus master

ATTACKING A MAC

STUNT HACK?!

- ▶ PoC - patch auth handler to bypass login screen
- ▶ Return success? Nah return 1 bro
- ▶ Log in with any password

```
__text:0000000000006334 loc_6334:                                ; CODE XREF: _ODRecordVerifyPassword+3C↑j
__text:0000000000006334                                ; _ODRecordVerifyPassword+58↑j ...
__text:0000000000006334                                mov     al, bl
__text:0000000000006336                                add     rsp, 68h
__text:000000000000633A                                pop     rbx
__text:000000000000633B                                pop     r12
__text:000000000000633D                                pop     r13
__text:000000000000633F                                pop     r14
__text:0000000000006341                                pop     r15
__text:0000000000006343                                pop     rbp
__text:0000000000006344                                retn
```




...today!
www.swissotel.com

00:0A:35:02:68:3F

Power

Transcend
CompactFlash®
ULTRA
2GB
Industrial

4199
U25R

1080063A

S/N 1150 - 9334
whizzsystems.com

SPARTAN-6
Spartan-6
Spartan-6
Spartan-6



FPGA development board



FPGA development board

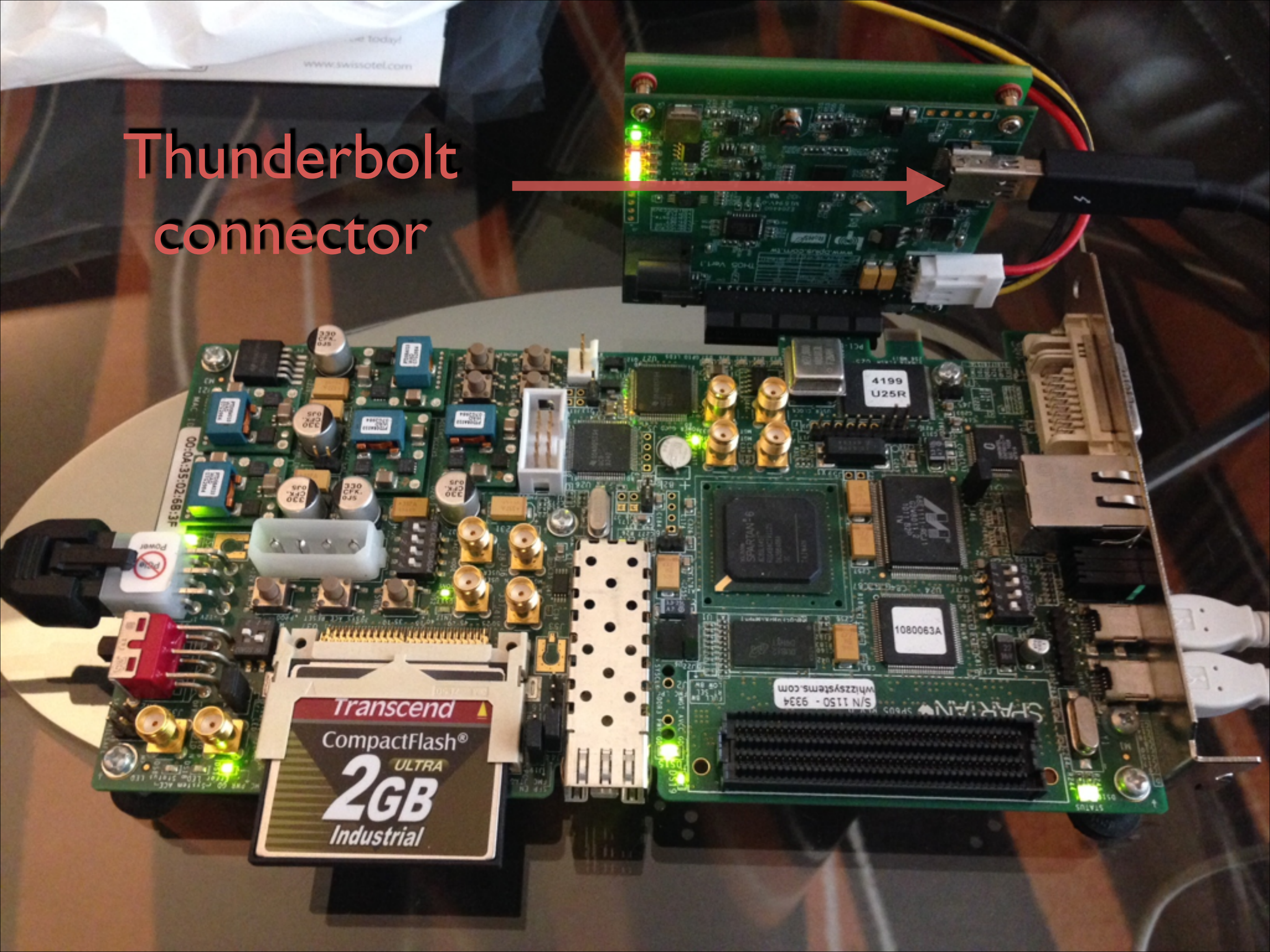
Thunderbolt to PCIe board

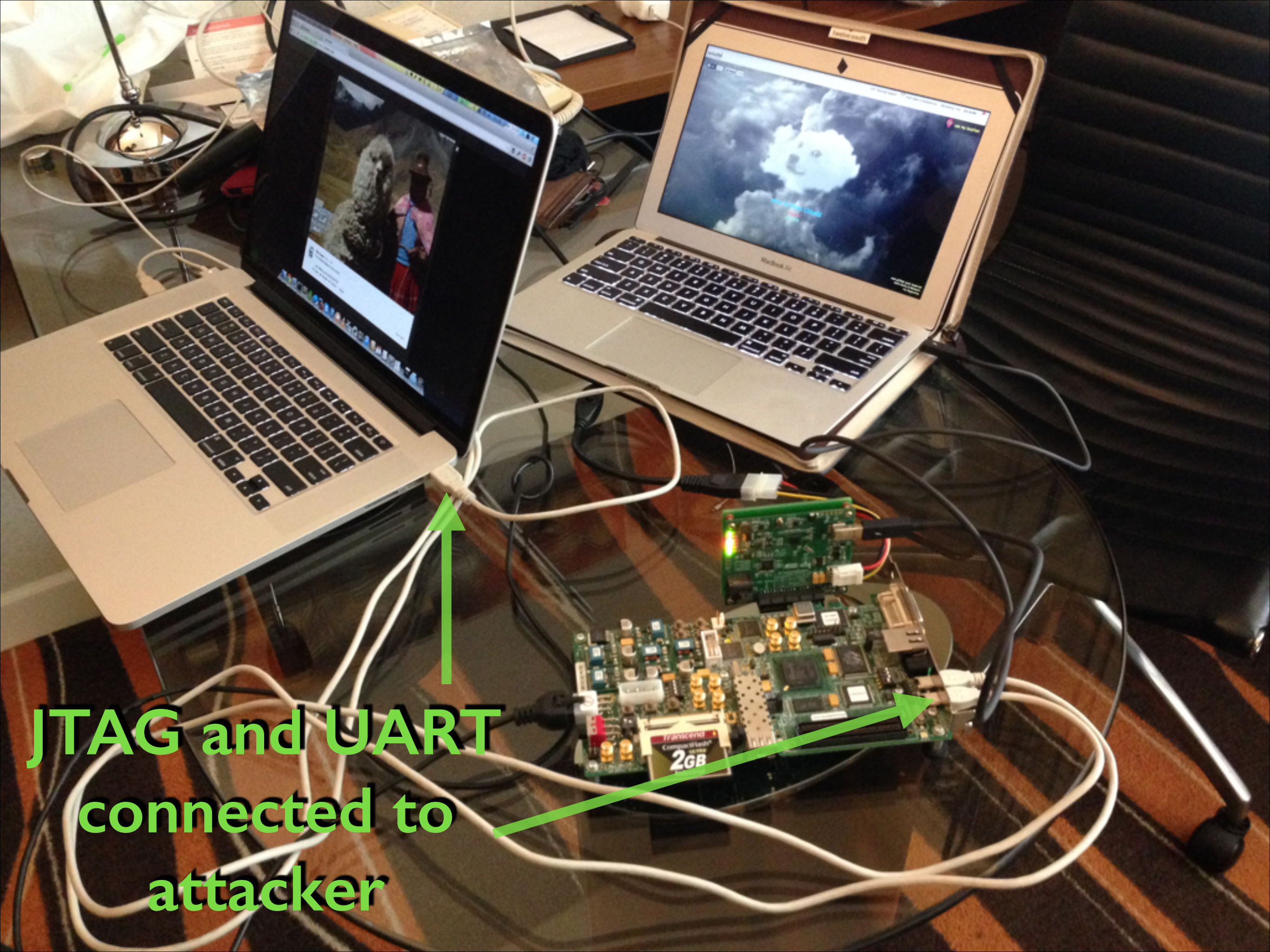


FPGA board
PCIe connector



Thunderbolt
connector

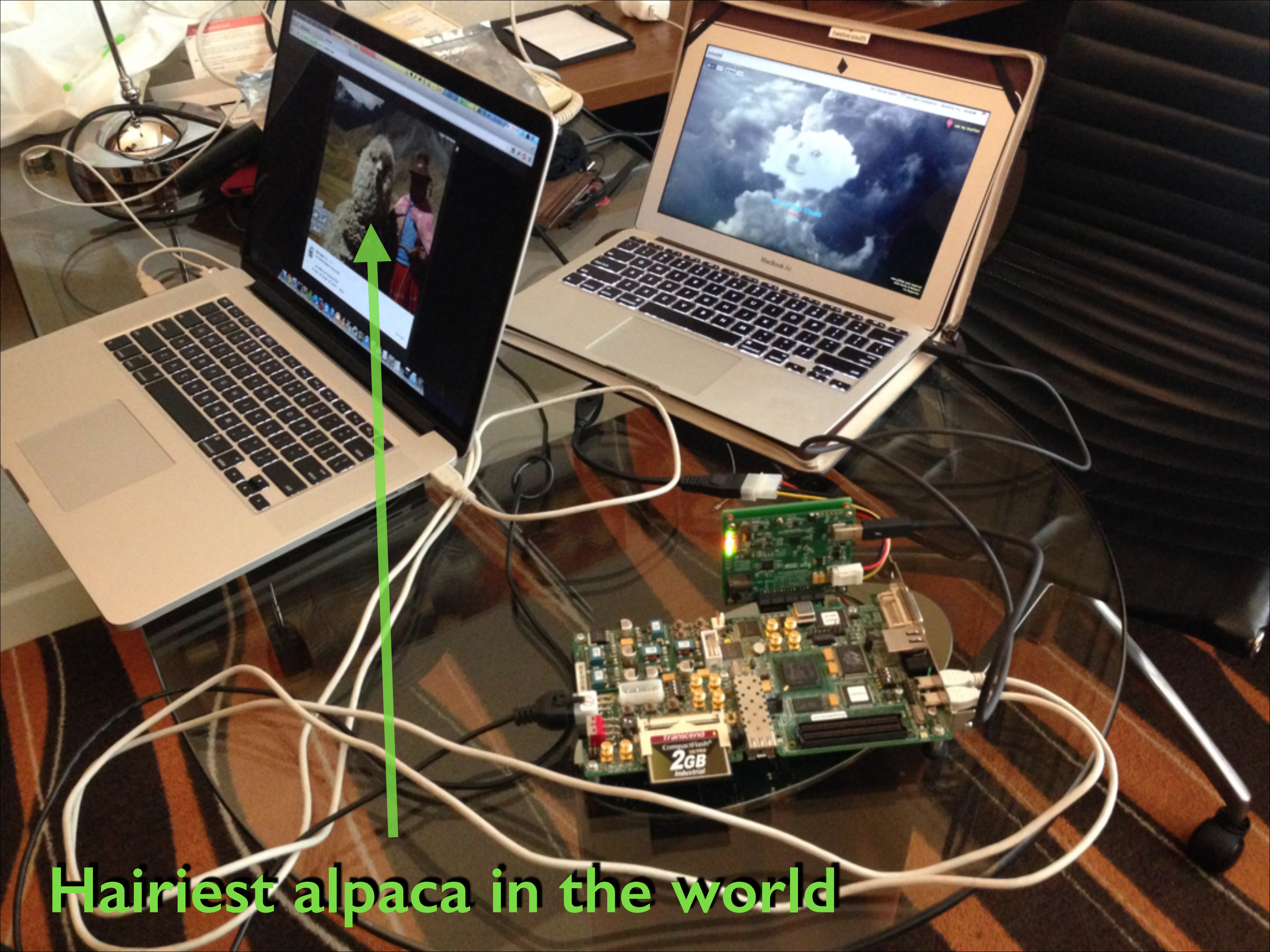




**JTAG and UART
connected to
attacker**



Thunderbolt connected to victim



Hairiest alpaca in the world

ATTACKING A MAC

STUNT HACK?!



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

ATTACKING A MAC

STUNT HACK?!



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

OTTERSTORM

JUST IN CASE OUR STUPID DEMO DIDN'T WORK



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

IT'S OK, WE MADE A VIDEO

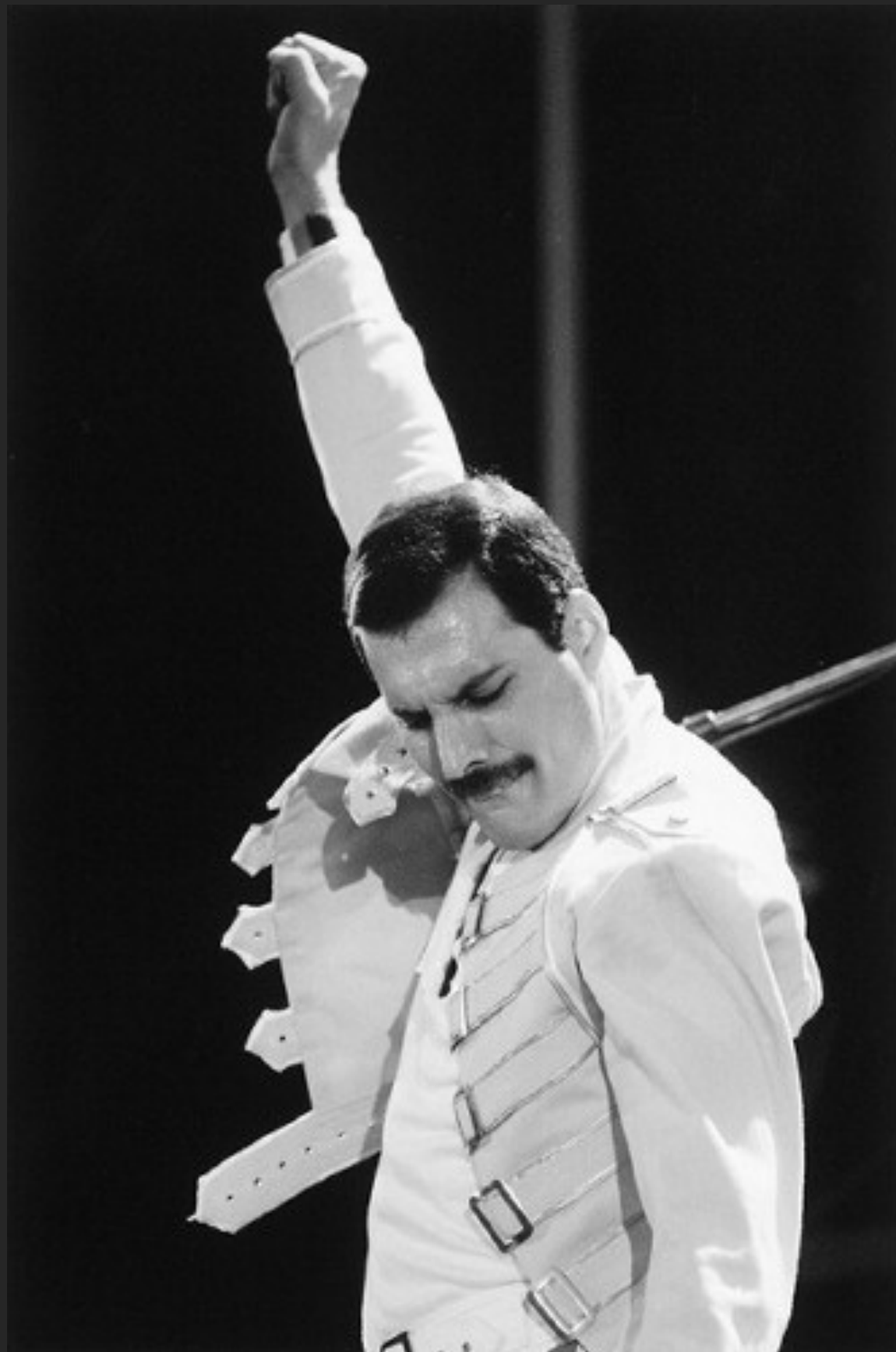


Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

IT'S OK, WE MADE A VIDEO



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening



**WAIT, THE DEMO
WORKED?**

YEP

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

THIS SEEMS BAD

Y'KNOW, IF YOU LIKE SECURITY AND STUFF

- ▶ Intel realised this was not a good “feature”
- ▶ What to do about it?
 - ▶ Glue all the ports shut?
 - ▶ Voodoo curse?
 - ▶ Access controls on device I/O?

THIS SEEMS BAD

Y'KNOW, IF YOU LIKE SECURITY AND STUFF

- ▶ Intel realised this was not a good “feature”
- ▶ What to do about it?
 - ▶ ~~Glue all the ports shut?~~
 - ▶ Voodoo curse?
 - ▶ Access controls on device I/O?

THIS SEEMS BAD

Y'KNOW, IF YOU LIKE SECURITY AND STUFF

- ▶ Intel realised this was not a good “feature”
- ▶ What to do about it?
 - ▶ ~~Glue all the ports shut?~~
 - ▶ ~~Voodoo curse?~~
 - ▶ Access controls on device I/O?

THIS SEEMS BAD

Y'KNOW, IF YOU LIKE SECURITY AND STUFF

- ▶ Intel realised this was not a good “feature”
- ▶ What to do about it?
 - ▶ ~~Glue all the ports shut?~~
 - ▶ ~~Voodoo curse?~~
 - ▶ Access controls on device I/O? 👍

VT=====D

INTEL, YOU BASTARDS

- ▶ Virtualised I/O
- ▶ Hypervisor can now assign devices directly to guests
 - ▶ This is how VMDirectPath works
- ▶ DMA requests are remapped w/access controls
- ▶ Interrupts are remapped w/access controls

VT=====D

INTEL, YOU BASTARDS

- ▶ VT-d unit has “domains”
- ▶ There is at least one domain (the host’s domain)
- ▶ In order to assign a device to a guest, the VMM creates a domain for that guest
 - ▶ Assigns a device to it

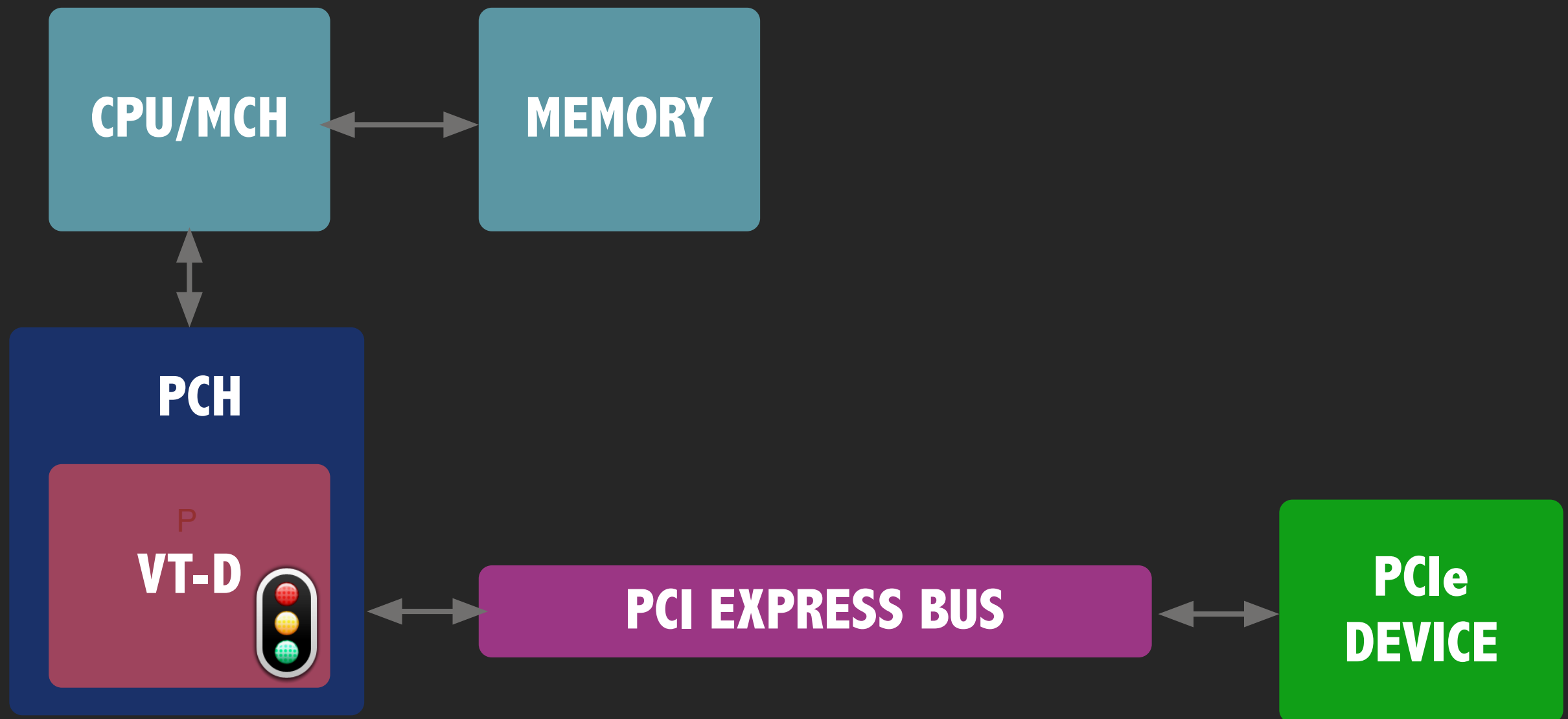
VT-D

A SECURITY FEATURE?

- ▶ OS X kernel configures VT-d
 - ▶ Actually it's the IOPCIFamily driver
 - ▶ All the devices are configured in a single VT-d “domain”
- ▶ Drivers allocate DMA buffers
 - ▶ New kernel memory allocator tells VT-d unit about regions
 - ▶ Now when DMA requests come in on the PCIe bus, VT-d says yea or nay
- ▶ If you are denied access, the kernel's VT-d handler is called and you see this in your console:
 - ▶ vtd[0] fault: device 0:20:0 reason 0x5 W:0x64c000

VT-D

INTEL, YOU BASTARDS



Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

VT-D

INTEL, YOU BASTARDS

- ▶ On all ≥ 2012 Macs (Ivy Bridge)
 - ▶ Requires OS config - supported in OS X since 10.8.2
- ▶ Restricts PCIe device DMA access
 - ▶ This is balls
 - ▶ Means our trix don't work on ≥ 2012 machine running $> 10.8.2$
- ▶ Windows pre-8 (AFAIK) doesn't configure VT-d
 - ▶ Pretty sure I remember reading that somewhere
- ▶ Linux does a much better job of configuring VT-d

AM I OWNED?

PROBABLY NOT

	<10.8.2	>=10.8.2
Pre-Ivy Bridge	OWNED	OWNED
Ivy Bridge and later	OWNED	NOT OWNED :(



DUDE, WHAT THE HELL? UPGRADE YOUR SHIT

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

WHAT'S NEXT?

NEW TRIX

- ▶ Maybe make the kit a little bit smaller
- ▶ Bypass VT-d?
- ▶ See if we can do it without imitating a device?
- ▶ Full memory capture

REFERENCES

- ▶ Metlstorm - Hit by a Bus (Ruxcon 2006)
 - ▶ http://www.security-assessment.com/files/presentations/ab_firewire_rux2k6-final.pdf
- ▶ Quinn the Eskimo - FireStarter (MacHack 2002)
 - ▶ <http://www.anarchistturtle.com/Quinn/WWW/Hacks.html>
- ▶ Inception (FireWire DMA tool)
 - ▶ <http://www.breaknenter.org/projects/inception/>
- ▶ PCIe Base Specification (507 pages, great night time reading)
 - ▶ http://read.pudn.com/downloads161/doc/729268/PCI_Express_Base_11.pdf
- ▶ Xilinx PCIe DMA Reference Design
 - ▶ http://www.xilinx.com/support/documentation/application_notes/xapp1052.pdf

Thunderbolts and Lightning ⚡⚡⚡ Very,Very Frightening

KTHXBAI

@scollinsonz
smc@affinity.net.nz
<http://affinity.net.nz>

@snare
snare@ho.ax
<http://ho.ax>
<http://blog.azimuthsecurity.com>

greetz:

vt, pipes, antic0de, quine, metlstorm, h l kar l, y0 l l, radian

special thanks to:

thomas motherfuckin' lim
statler and waldorf (nagy and grugq)

mad props to:

barns. now let's get grimy.